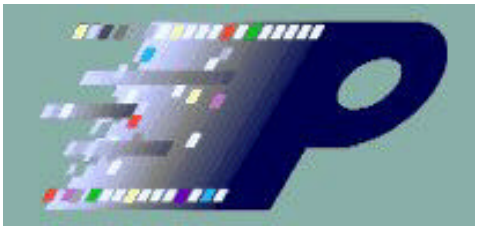

DPhyPreprocCtl User's Manual

Table of Contents

1 Overview 1

2 Setup & Installation 2

3 DPhy Preprocessor Concepts 4

3.1 Records 4

3.2 Filtering 4

3.2.1 LP Record filtering 4

3.2.1.1 Delta Mode 4

3.2.1.2 Escape Mode 5

3.2.2 Packet filtering 5

3.3 Triggering 5

4 Using DPhyPreprocCtl 7

4.1 Connecting to the Instrument 7

4.2 Main Window Controls 7

4.2.1 Configuration Controls 8

4.2.2 Filter Controls 10

4.2.3 Trigger Controls 11

4.2.3.1 Packet Fields Triggering 11

4.2.3.2 LP Signal State Triggering 14

4.2.3.3 LP Sequence State Triggering 15

4.2.3.4 Escape Command Triggering 15

4.2.3.5 CRC Error Triggering 16

4.2.3.6 ECC Error Triggering 16

4.2.4 Status Controls 16

4.2.4.1 Activity 16

4.2.4.2 Statistics 16

4.2.4.3 Statistics Control 18

4.3 Defining Custom Commands 18

4.4 Firmware Update 19

4.5 Configuration Settings 20

4.6 Menus 20

Contacting The *Moving Pixel* Company

Phone	+1.503.626.9663	US Pacific Time Zone
Fax	+1.503.626.9653	US Pacific Time Zone
Address	**The *Moving Pixel* Company** 4905 SW Griffith Drive, Suite 106 Beaverton, Oregon 97005 USA	
Email	information@movingpixel.com	
Web site	http://www.movingpixel.com	
Documentation		

1 Overview

The Moving Pixel Company DPhy Preprocessor is an instrument designed to monitor traffic on up to 4 DPhy lanes of the MIPI standard.running up to 1.5 Gbps. The instrument operates as a preprocessing front-end to a Tektronix Logic Analyzer (TLA) for the acquisition and decoding of DPhy bus traffic.

The DPhy Preprocessor's main functions are to:

- Process DPhy data and package into records for TLA acquisition
- Convert serial HS packet data to parallel bytes
- Convert encoded LPDT and escape mode data to parallel bytes
- Provide sophisticated protocol and state triggering for the TLA
- Provide DSI/CSI packet filtering based on packet type
- Provide link activity indicators, status, and statistics

The DPhy Preprocessor is controlled via a Windows application called DPhyPreprocCtl, which communicates with the preprocessor via a USB connection. Using the GUI of this application, the user can configure the triggering and filtering functions of the instrument. This document describes the installation and operation of DPhyPreprocCtl.

2 Setup & Installation

This section describes the steps for installing DPhyPreprocCtl on the host computer, which may be the TLA but this is not required. The host computer is expected to be a Windows machine running Windows XP, but the application can run under Vista or Windows 7 (32-bit) with the appropriate driver. Please contact The Moving Pixel Company for support in using these alternate operating systems.

To install DPhyPreprocCtl, simply execute the setup.exe file provided and step through the installation windows. The DPhyPreprocCtl application is installed by default in the **c:\Program Files\TMPC\DPhyPreprocCtl** directory.

One requirement for running the DPhyPreprocCtl application is the Microsoft .NET Framework 3.5 platform. You can check if .NET 3.5 is installed through **Start->Settings->Control Panel** and double-clicking "Add or Remove Programs". Scroll through the list of applications for "Microsoft .NET Framework 3.5".

If you are installing from an installation CD provided by The Moving Pixel Company, Microsoft .NET 3.5 will be installed automatically (if necessary). On the other hand, if you are running the application installation from a setup file downloaded from our website or received via email, the setup files for installing .NET 3.5 will not be present.

In this case, running the setup.exe on a machine without .NET 3.5 installed will appear to begin .NET installation (asking the user to accept supplemental license terms), but then will subsequently report an error installing system components. If this occurs, the user should close the error report and manually install .NET 3.5.

To install Microsoft .NET 3.5 (without an installation CD from The Moving Pixel Company) your computer must be connected to the internet. The following is a link to the installation executable (note: this file is over 200 MB):

http://download.microsoft.com/download/6/0/f/60fc5854-3cb8-4892-b6db-bd4f42510f28/dotnetfx35.exe

Download the executable and run it and follow the setup instructions.

Finally, it has been discovered that .NET 3.5 installation may fail on some machines when the Windows IIS service is installed. It appears that some TLAs have this service installed by default even though it is not necessary for normal operation (the Windows IIS service is in support of making your computer a web server). Thus, if you cannot successfully install .NET 3.5, it may be necessary to uninstall the Windows IIS service before trying again. Please note that if re-installation of the Windows IIS service after .NET 3.5 installation is desired, this will require a Windows XP System CD.

To uninstall the Windows IIS service, do the following:

1) Select Start->Settings->Control Panel
2) Double-click "Add or Remove Programs"
3) Click on "Add/Remove Windows Components" in left pane
4) Uncheck the "Internet Information Service (IIS)" box
5) Click Next button
6) Click Finish button

Please be aware that if .NET 3.5 is not installed on your machine, DPhyPreprocCtl will NOT run at all (nothing will happen at launch -- no windows, warnings, or errors will appear).

3 DPhy Preprocessor Concepts

3.1 Records

The main function of the DPhy Preprocessor hardware is to monitor up to four lanes of DPhy bus traffic and package data into records for TLA acquisition and display. All facets of bus communication are captured from lane LP signaling to escape mode sequences to LPDT and HS burst packets.

Depending on the current DPhy bus state, one of two types of records are output to the TLA. While the exact details of the two record types is not necessary for the user to understand, a quick overview may be helpful.

If one or more lanes are transmitting data in an HS burst, then an HS data record is output. Otherwise, an LP data record is output. Both record types have associated status that conveys auxiliary state information about the record including various header and CRC tags, trigger and filtering information, etc.

An HS data record contains information about each lane: either an HS data byte if the lane is currently transmitting an HS packet, or the LP state of the lane if it is not (normally this would be the LP11 state). An LP data record contains information about the LP state of all lanes (including the clock lane) as well as possibly a decoded LP data byte from an escape command. In addition, the LP data record contains what is called the lane zero "sequence state", which is an encoded value representing a state in the LP signaling state machine (e.g. BTA).

3.2 Filtering

Another function of the DPhy Preprocessor is to filter records based on certain user criteria, restricting the data acquired by the TLA so that it contains more useful information to the user. The preprocessor implements two filtering mechanisms:

1) LP record filtering based on mode setting ("Delta" and/or "Escape")
2) DSI/CSI packet filtering based on user criteria

The next two sections describe these mechanisms.

3.2.1 LP Record filtering

The LP record filtering modes are extremely useful -- the user is likely to always enable them except for very low-level debugging acquisitions.

3.2.1.1 Delta Mode

One method for filtering LP records is called Delta Mode. When this mode is enabled, only records containing a lane that has changed state from the previous record is output. This significantly compresses the record set captured by the TLA and compacts the

disassembly view. As timestamps can indicate duration in a particular LP state, duplicated records are superfluous.

For example, much of the time on the DPhy bus is spent with all lanes in LP11. In delta mode, all such periods will be represented by a single record. Another example is in viewing any LP signaling (HS burst, BTA, Escape, etc). Generally, signal states in these modes persist for one clock period of the LP frequency, which would result tens to hundreds of duplicated records in the TLA. In delta mode, these record sets would be reduced to one record.

An enhancement to delta mode is the option to specify a timeout to force output of periodic records even though no change has occurred. This allows the TLA to finish acquisition even if the bus is static for a long period of time.

3.2.1.2 Escape Mode

A second method for filtering LP records is called Escape Mode. This mode is used to filter records that represent bit transitions in escape mode, allowing only those that occur on packet byte boundaries to be passed to the TLA. Because of the Spaced-One-Hot coding method used for communication in escape mode, this will increases useful record density by 16x. *Note that if Escape Mode is not set, no decoding of escape bytes is performed.* The implication of this is that no filtering or triggering of LPDT packets is performed.

3.2.2 Packet filtering

Another filtering mechanism of the instrument is the ability to filter MIPI packets based on packet type, simplifying the record trace, eliminating records of no interest to the user. This function can also be viewed in the opposite sense as that of *selecting* records of a particular packet type. Using this feature implements focused acquisition to view only specific records of interest.

Packet filtering is applied based on the DataType field in the packet header and applies to both HS and LPDT packets. DCS packets can also be filtered using the DCSCmd byte.

Implementation note: in the case of LP record filtering, filtered records are simply not output. However, for packet filtering, HS records may contain up to 4 lanes and packets can start and end in the middle of a HS record. In this case, information is contained in the data record indicating which lanes should be filtered allowing the TLA to display data accordingly.

3.3 Triggering

Triggering, as implemented by the preprocessor, is distinct from triggering from the perspective of the TLA. Often they set up to be synonymous such that when the preprocessor triggers, so does the TLA, but this is not required. The preprocessor simply sets a trigger flag in records output to the TLA that satisfy its trigger criteria, that is

qualifying information specified by the user though the DPhyPreprocCtl GUI. There is no restriction on the number of records that may have their trigger flag set.

How this field is used in the trigger state machine on the TLA is outside the scope of this document. In the simplest and most common case, the TLA is set to trigger when the trigger flag is true.

In the DPhy Preprocessor, various trigger modes are supported. With the exception of the trigger modes for CRC or ECC errors, the remaining modes require the user to define predicate terms that are then used in an arbitrary Boolean trigger equation. Often this equation is simply and ANDing or ORing of predicate terms.

The most common form of triggering is based on packet data, i.e. predicate terms are built from comparing masked fields in DSI/CSI packets are compared to constants. Other trigger modes use LP Signal State, lane zero LP Sequence State (e.g. BTA), or the lane zero Escape command to build their predicates.

4 Using DPhyPreprocCtl

4.1 Connecting to the Instrument

When the application is started, a connection dialog (see Figure 1) is shown for the user to select a DPhy Preprocessor instrument powered-on and reachable via USB. This dialog can also be brought up for the user to switch instruments using the option in the Control menu labeled "DPhy Preprocessor Connect...".

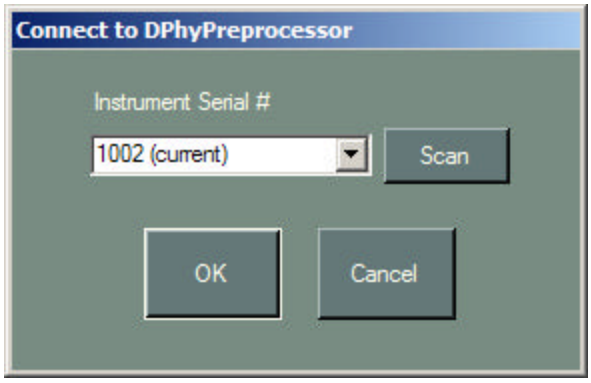

Figure 1 – Connection Dialog

A drop-down box is provided with reachable instrument serial numbers as well as an option called "Offline". Offline mode allows using the application to be used without a connection to real hardware, useful for demonstration of the GUI and building and saving configurations. There is also a "Scan" button that the user can press to request that the application search for any new instruments that have recently been powered on.

Note that while multiple copies of the application can run at the same time, only one instance can be connected to a given instrument at a time. Serial numbers of instruments that are in-use by another copy of the application are marked "(in use)".

To connect to an instrument, simple select the appropriate serial number and click OK. Note that the serial number and firmware version of the currently connected instrument appears in the status bar at the bottom right of the main window.

4.2 Main Window Controls

Once connected, the main window is shown (see Figure 2). The instrument serial number is shown in a bottom right pane on the status bar. Except for minor support dialogs accessed through the menus, the main window contains all the user-controls for configuring and operating the instrument.

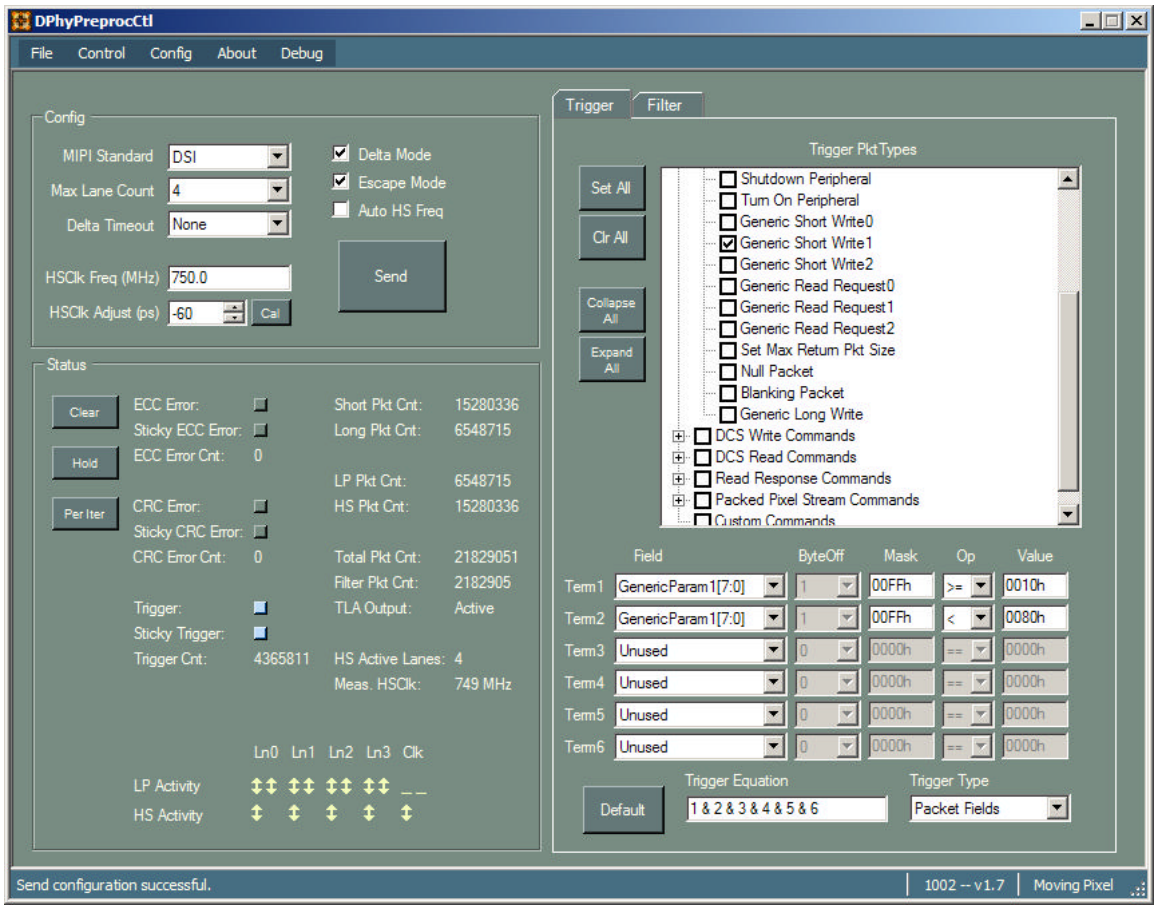

Figure 2 – DPhyPreprocCtl GUI

The main window is divided into four areas of controls: configuration, status, trigger control, and filter control (the last two share a tabbed page) The following sections describe each of these areas.

4.2.1 Configuration Controls

The section of the main window provides controls that support overall mode configuration and are located in the upper-left section of the main window. They are:

MIPI Standard – drop-down control to select the DSI or CSI standard. This selection should be made first as it naturally resets the filtering and triggering options available to the user.

Max Lane Count – drop-down control to select the maximum number of lanes to monitor for activity. Unused lanes are forced to LP11 for the purposes of instrument monitoring and acquisition. Note that the instrument DPhy bus activity less than or equal to the Max Lane Count setting. For example, with a Max Lane Count setting of 4, all HS bursts using 1, 2, 3, or 4 lanes will be detected automatically and the actual number of lanes participating in ongoing HS bursts is displayed in the status region.

Delta Mode – check box to enable delta mode filtering, which restricts record output to records that reflect a state change since the last record. See section 3.2.1.1 for more details.

Delta Timeout – drop-down control to set the delta mode time out duration. This setting is the maximum time to wait without outputting a record before forcing a record output. If timeout is set to "None", the instrument will wait indefinitely for a change in inputs.

Escape Mode – check box to enable escape mode filtering, which restricts LP record output during escape sequences to those that contain a decoded byte (as opposed to intermediate bits). See section 3.2.1.2 for more details.

Note that Escape Mode is a required setting if the user is to trigger/filter based on packets sent in LPDT mode. If Escape Mode is not set, records that contain the decoded LPDT byte will be interleaved with intermediate records and packet triggering/filtering will not apply to these records.

HSClk Freq – text box for the user to indicate the approximate lane clock frequency. This value is used to set the sampling phase of the data relative to the clock on input. Its accuracy becomes more important with increasing frequency.

Auto HS Freq – check box to enable the instrument to automatically detect the HS clock frequency. When checked, the HSClk Freq text box is disabled and automatically filled in with the detected HSClk frequency.

This mode is very useful. However, HSClk frequency analysis requires a constant clock for a significant period (milliseconds) during measurement. Thus, this mode is not suitable for acquisitions in which the clock turns on and off while the instrument is monitoring the link. Also, when this mode is enabled, note that when the clock frequency changes there may be a brief period when errors are detected before the frequency change is detected. Thus, as a convenience, all statistics are reset when a significant clock change is detected.

HSClk Adjust – spin control to adjust fine delay timing on the incoming HSClk for very high-frequency acquisitions (i.e. 600-750 MHz). This timing control adjusts the clock *relative to the ideal quadrature setting* for acquisition and, normally, this value should thus be set to zero. However, modest adjustments are sometimes required depending on less-than-ideal acquisition conditions depending on input signal integrity, non-standard volatages, cabling, noise, or timing skew on the bus.

Cal – button to automate HSClk adjustment. To use, set up the system so valid HS traffic is continuously sent on the bus. Then click this button to initiate a sweep search within +/- 400 ps around zero in 20 ps increments, measuring the number of ECC and CRC errors received in a short time period. The sweep takes a few seconds and when complete, the optimum HSClk adjustment setting is displayed and assigned to the control.

Send – button to send the current configuration (including filter and trigger settings) to the preprocessor. This is required operation to update changes made in the triggering or filtering configuration. The user is notified that Send should be clicked by the appearance of the text "Modified" to the lower-right of the Send button. Note that changes to settings described in this section – configuration controls – are sent immediately to the DPhyPreprocessor and do not require a Send button click.

4.2.2 Filter Controls

The filter control page is one of two tab pages occupying the right side of the main window. To show the page, click on the Filter tab at the top.

The central feature of the filter control section is a checked tree-view containing all packet types. Top-level nodes in the tree-view consist of categories of packet types and second-level nodes in the tree-view represent the packet types themselves. Category nodes are not able to be checked by the user, but they are automatically checked if at least one node within the category is checked.

To select a packet type to be filtered, the user simply opens the tree node category (if closed) and checks the packet type to be filtered by the preprocessor (and then clicks the Send button). When a packet it filtered by the DPhyPreprocessor its data bytes are tagged with a filter flag in the output record stream to the TLA. This allows the TLA to remove it from the displayed packet listing.

For DSI, the top-level categories are:

- Generic Commands
- DCS Write Commands
- DCS Read Commands
- Read Response Commands
- Packed Pixel Stream Commands
- Custom Commands

For CSI, the top-level categories are:

- Generic Commands
- YUV Commands
- RGB Commands
- Raw Commands
- User Commands
- Custom Commands

These categories are mostly self-explanatory, based on the packet type definitions for each respective MIPI specification. The Generic Commands category includes all packet types defined in the specification that do not fall in any of the other categories. The Custom Commands category includes user-defined data types using the Custom Command dialog (Config -> Define Custom Commands… menu option).

Other controls in this section are:

Set All – button that checks all the packet types in the tree-view

Clr All – button that clears all the checked packet types in the tree-view

Collapse All – button the collapses all the category nodes in the tree-view

Expand All – button that expands all the category nodes in the tree-view

4.2.3 Trigger Controls

The trigger control page is one of two tab pages occupying the right side of the main window. To show the page, click on the Trigger tab at the top.

As outlined in section 3.3, the DPhy preprocessor has several triggering modes. The triggering mode is selected via the Trigger Type drop-down control at the bottom-right corner of the Trigger Controls section. The trigger mode should be selected first before other fields as the control options change based on mode. The trigger modes available are:

- **Packet Fields** – This is the most common form of triggering. Comparisons of packet types and packet fields are used to create predicates evaluated in a Boolean equation to determine trigger state.
- **LP Signal State** – This mode uses comparisons of LP signal state for any or all lanes (including the clock lane) as predicates evaluated in a Boolean equation to determine trigger state.
- **LP Sequence State** – This mode uses comparisons of the lane zero sequence state (e.g. BTA) as predicates evaluated in a Boolean equation to determine trigger state.
- **Escape Command** – This mode uses comparisons the escape command byte as predicates evaluated in a Boolean equation to determine trigger state.
- **CRC Error** – This mode sets the trigger flag on CRC error..
- **ECC Error** – This mode sets the trigger flag on ECC error.

4.2.3.1 Packet Fields Triggering

When triggering based on packet fields, triggering is a two stage process. The first stage selects the set of packets based on packet type for further evaluation.[1] The second stage then evaluates these qualifying packets using the designated predicate terms and trigger equation to determine the trigger flag state.

[1] Note that both HS packets and LPDT packets can be considered for triggering in this mode, but only if Escape Mode is enabled. Otherwise only HS packets are considered.

4.2.3.1.1 Stage 1 Trigger Selection

The first stage selection process of packets for trigger evaluation is identical to packet filtering for output. As with the filter control section, the trigger control section has a checked tree-view containing all packet types. Top-level nodes in the tree-view consist of categories of packet types and second-level nodes in the tree-view represent the packet types themselves. Category nodes are not able to be checked by the user, but they are automatically checked if at least one node within the category is checked.

To select a packet type to be included in trigger evaluation, the user simply opens the tree node category (if closed) and checks the desired packet type. Note that in this mode, if no packets are selected, *no trigger will result*, regardless of the trigger equation settings.

For DSI, the top-level categories are:

- Generic Commands
- DCS Write Commands
- DCS Read Commands
- Read Response Commands
- Packed Pixel Stream Commands
- Custom Commands

For CSI, the top-level categories are:

- Generic Commands
- YUV Commands
- RGB Commands
- Raw Commands
- User Commands
- Custom Commands

These categories are mostly self-explanatory, based on the packet type definitions for each respective MIPI specification. The Generic Commands category includes all packet types defined in the specification that do not fall in any of the other categories. The Custom Commands category includes user-defined data types using the Custom Command dialog (Config -> Define Custom Commands… menu option).

Other controls in this section are:

Set All – button that checks all the packet types in the tree-view

Clr All – button that clears all the checked packet types in the tree-view

Collapse All – button the collapses all the category nodes in the tree-view

Expand All – button that expands all the category nodes in the tree-view

4.2.3.1.2 Stage 2 Trigger Selection

Stage two trigger selection defines the predicates and the trigger equation to be used for final trigger evaluation. Six predicate terms are supported where each term targets a 16-

bit field (byte aligned) anywhere in packet bytes 1-11. This byte range allow all fields in short packets and long DCS packets with up to three 16-bit arguments to be fully supported. Predicates are defined by masking the field and comparing it to a value. After the predicate terms have been defined, the user specifies a logical trigger equation that uses the field comparison results.

The following list describes the controls used for term definitions:

- **Field** – drop-down box identifying the field name. This use of this control is not required, as it simply acts as a "preset" for other controls in the term. However, if this field is used, it should be selected first, as it will overwrite the other term controls with appropriate values.

  The first option is always "Unused" which sets the term controls to an always true comparison. This allows it to be included in the default trigger equation (all terms ANDed) with no effect. This selection is merely a convenience for readability because the trigger equation actually specifies whether a term is used or not and, thus, it is not necessary to set unused terms to this setting.

  The second option in the field drop-down control is always "Custom", which enables all the other term controls for user settings.

  Next, follow common generic packet header fields (i.e. VC, DataType, DataID, WordCount, ECC) which initialize the byte offset and mask fields appropriately.

  Finally, any additional options vary with the selected packet type. If a single packet type with known fields is selected in stage 1, then they are included as options here.

- ByteOff – drop-down box to set the byte offset of the field. For example, a byte offset of 4 would specify the field to be the first two payload bytes of a packet. (Note in this case that all of the selected packets from stage 1 should be long packets as byte 4 is not defined for short packets.).
- Mask – text box to set the mask value. This value is treated as a 16-bit value and ANDed with the packet field before comparison with the term value. Values are parsed as decimal unless appended with an 'h', in which case they are parsed as hexadecimal (i.e. 20h = 32 decimal). The LSB corresponds to the earlier byte in the packet. For example, if the mask were set to FF00h with ByteOff == 4, the resulting masked value would have packet byte 5 in the MSB and zero in the LSB.

- **Op** – drop-down box to select the predicate comparison operation. All logical comparisons are supported, i.e. ==, !=, >, >=, <, <=.
- **Value** – text-box to set the term comparison value. Values are parsed as 16-bit decimal numbers unless appended with an 'h', in which case they are parsed as

16-bit hexadecimal (i.e. 20h = 32 decimal). As with the mask value, the LSB represents the earlier packet byte.

The final controls used for stage two triggering are the trigger equation specification itself and the default button:

- **Trigger Equation** – text box to specify how term predicates should be combined in a Boolean equation to determine the final trigger state. Terms predicates are identified using the numbers '1' through '6'. The '&' and '|' symbols for AND and OR operations and parentheses can be used to set precedence (e.g. "1 & (2 | 3) & 4").
- **Default** – button to reset all term controls and the trigger equation to their default. The default setting is for all term controls to be "Unused" and the trigger equation to be "1 & 2 & 3 & 4 & 5 & 6".

4.2.3.2 LP Signal State Triggering

When triggering based on LP Signal State, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition. [2] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate on a composite value derived from LP records. This composite value is 10 bits and indicates the current LP state of all lanes including the clock lane. The five 2-bit fields represent the LP state of each lane, where 0 = LP00, 1 == LP01, 2 == LP10, and 3 == LP11.

Bit definitions for the fields are as follows:

- [1:0] lane 0 LP state
- [3:2] lane 1 LP state
- [5:4] lane 2 LP state
- [7:6] lane 3 LP state
- [9:8] clock LP state

For example, a mask of 0x300 and a value of 0x100 define a predicate to test if the clock lane is in the LP01 state.

For convenience, the field drop-down box in this mode contains entries for all combinations of lane and state, i.e. "Lane0 in LP00", "Lane0 in LP01"... "Clock in LP10", "Clock in LP11". Also, note that the ByteOff field is unused in this mode and is disabled.

[2] Note that in this mode, triggering is disabled for the special case where all terms are set to "Unused, regardless of the trigger equation.

4.2.3.3 LP Sequence State Triggering

When triggering based on LP Sequence State, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition.[2] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate solely on the lane zero sequence state. DPhy defines various LP signaling sequences such as BTA, ESCAPE, and HS and triggering is possible on the occurrence of any of them. While each state and intermediate states are represented by a particular field value, only a few are generally useful. These particular states – "HS Burst", "BTA", "BTA Exit" and "Escape Entry" – are available for convenience in the field drop-down box.

For completeness, Table 1 documents all the possible sequence states. Numbers in the state names indicate receipt of a particular LP signal value (i.e. 0 = LP00, 1 = LP01, 2 = LP10, 3 = LP11). For example, if the sequence state is LPSTATE_3, lane 0 is currently in the LP11 state. If in LPSTATE_32, lane 0 has seen LP11 followed by LP10.

Table 1 – Lane 0 Sequence States

State	Value
LPSTATE_UNKNOWN	0
LPSTATE_3	1
LPSTATE_32	2
LPSTATE_320	3
LPSTATE_3202	4
LPSTATE_BTA	5
LPSTATE_BTA_EXIT	6
LPSTATE_3201	7
LPSTATE_ESCAPE	10
LPSTATE_31	11
LPSTATE_HS	12

4.2.3.4 Escape Command Triggering

When triggering based on the Escape Command, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition.[2] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate on the escape command value.

For convenience, the field drop-down box in this mode contains entries for predefined escape commands in the DPhy specification. These are: "LPDT", "ULPS", "Reset Trigger", "Trigger 1", "Trigger 2", and "Trigger 3". The user may also select "Custom Escape Cmd" to define other command values.

.

4.2.3.5 CRC Error Triggering

When triggering on CRC errors, all other trigger controls are disabled. The trigger flag is set for packets that have a CRC error.

4.2.3.6 ECC Error Triggering

When triggering on ECC errors, all other trigger controls are disabled. The trigger flag is set for packets that have an ECC error.

4.2.4 Status Controls

When connected to a DPhyProcessor instrument, the status control section provides continuously updated statistics about bus activity as monitored by the preprocessor. Status updates occur about once a second.

The status display generally falls into three categories: activity, statistics, and statistics control.

4.2.4.1 Activity

The activity display at the bottom of the status pane shows graphically the LP and HS activity occurring on each lane. The following notes describe the activity display:

- Activity for the LP+, LP-, and HS signals or each lane is represented by an indicator graphic.
- The top row for LP activity consists of 10 indicators, a pair for each lane, with the left indicator in each pair representing LP+ and the right indicator representing LP-.
- The bottom row for HS activity consists of 5 indicators, one for each lane.
- Indicators can be low (low horizontal bar), high (high horizontal bar), or transitioning (vertical bar with arrows on both ends).
- The indication spans the period since the last status update, so for example, if a toggling indicator appears, it means that the signal transitioned at least once since the last update.
- Indicators are valid and updated even if the acquisition pipeline is stalled waiting for an event (i.e. TLA Output is inactive -- for example, while in delta mode with no timeout and a quiescent bus).

4.2.4.2 Statistics

Many statistic about data traffic are collected and displayed in this section, the predominant part of the status control display. Each statistic is summarized below:

The ECC byte in packet headers check the integrity of the packet header. The ECC is tested for packets sent in both HS and LPDT mode if Escape Mode is enabled. Otherwise, the ECC is tested (and errors can be triggered on) only in HS packets. Controls and statistics relating to ECC status are:

- **ECC Error** – LED-control indicating whether an ECC error occurred since the last update. If red, an error occurred; if green, no error occurred.
- **Sticky ECC Error** – LED-control indicated whether an ECC error occurred since the last clear. If red, an error occurred; if green, no error occurred.
- **ECC Error Cnt** – counts the number of ECC errors that occurred since the last clear.[3]

The CRC bytes at the end of long packets check the integrity of the packet payload for long packets. The CRC is tested for packets sent in both HS and LPDT mode if Escape Mode is enabled. Otherwise, the CRC is tested (and errors can be triggered on) only in HS packets. Controls and statistics relating to CRC status are:

- **CRC Error** – LED-control indicating whether a CRC error occurred since the last update. If red, an error occurred; if green, no error occurred.
- **Sticky CRC Error** – LED-control indicated whether a CRC error occurred since the last clear. If red, an error occurred; if green, no error occurred.
- **CRC Error Cnt** – counts the number of CRC errors that occurred since the last clear.[3]

The trigger flag is set whenever the conditions defined by the trigger controls are met. Controls and statistics relating to trigger status are:

- **Trigger** – LED-control indicating whether a trigger event occurred since the last update. If light blue, a trigger occurred; if green, no trigger occurred.
- **Sticky Trigger** – LED-control indicated whether a trigger event occurred since the last clear . If light blue, a trigger event occurred; if green, no trigger occurred.
- **Trigger Cnt** – counts the number of trigger events that occurred since the last clear.[3]

Packet count statistics summarize the number of occurrences of each type of packet.[4] Statistics relating to packet counts are:

- **Short Pkt Cnt** – counts the number of short packets since the last clear.
- **Long Pkt Cnt** – counts the number of long packets since the last clear.
- **LP Pkt Cnt** – counts the number of LS packets since the last clear.
- **HS Pkt Cnt** – counts the number of HS packets since the last clear.
- **Total Pkt Cnt** – counts the total number of packets since the last clear.

[3] Note when any of the ECC, CRC, or Trigger counts reach 999 million, they are all reset to zero and continue counting.
[4] Note when any of these packet counts reach 999 million, the are all reset to zero and continue counting.

- **Filter Pkt Cnt** – counts the number of packets filtered since the last clear.

Lastly, other miscellaneous statistics provided are:

- **TLA Output** – displays "Active" if valid records are being output to the TLA and "Inactive" otherwise. Two common cases when the TLAOutput will be inactive are when in Delta Timeout mode with no timeout and a quiescent bus, and when in Escape mode when the Escape entry sequence has been seen but no transitions are occurring, in particular, when the lane is in ULPS.
- **HS Active Lanes** – Indicates the maximum number of lanes that were part of an active HS burst since the last clear button reset. If zero, no HS packets were seen.
- **Meas. HSClk** – Indicates the measured HS clock over a short time period (milliseconds). Note that this value is correct only if the clock does not turn on and off during the measurement period. Otherwise, the value displayed represents the average clock frequency.

4.2.4.3 Statistics Control

Three buttons control how statistics are displayed:

- **Clear** – resets all statistics to zero. The clear function is also automatically applied when the Send button is pressed or when in Auto HS Freq and the detected frequency changes by more than 10 MHz.
- **Hold / Resume** – suspends or resumes statistics update. The button label alternates with each click, indicating the function to be performed when next pressed. For example, when "Resume" is displayed, statistics update is halted until the button is pressed.
- **Totals / Per Iter** – shows total counts or per update counts. The button label alternates with each click, indicating the function to be performed when next pressed. For example, when "Totals" is displayed, all statistics except for the sticky errors and trigger, summarize activity only since the last update (rather than since the last clear).

4.3 Defining Custom Commands

Custom commands are DataTypes or DCS command codes that are not defined in the relevant MIPI specification. If present in the DPhy bus traffic, custom DataTypes *for long packets* are important to enumerate to ensure the DPhyPreprocessor does not get confused during parsing. This is because the default behavior when the instrument encounters an unknown DataType is to assume it is a short packet. In addition, if the user wants to filter or trigger on fields particular to a custom packet type, these DataTypes and/or DCS command codes need to be enumerated as well.

To define custom commands, a dialog is provided when "Define Custom Commands…" is selected from the Config menu (see Figure 3). This dialog lets you specify custom

short packet data types, long packet data types, and DCS commands. Simply enter lists of values (separated by spaces) in the appropriate text boxes and click OK. Note that DataTypes do not include the VC field and are masked to 6-bits. Values are parsed as decimal unless appended with an 'h', in which case they are parsed as hexadecimal (i.e. 20h = 32 decimal).

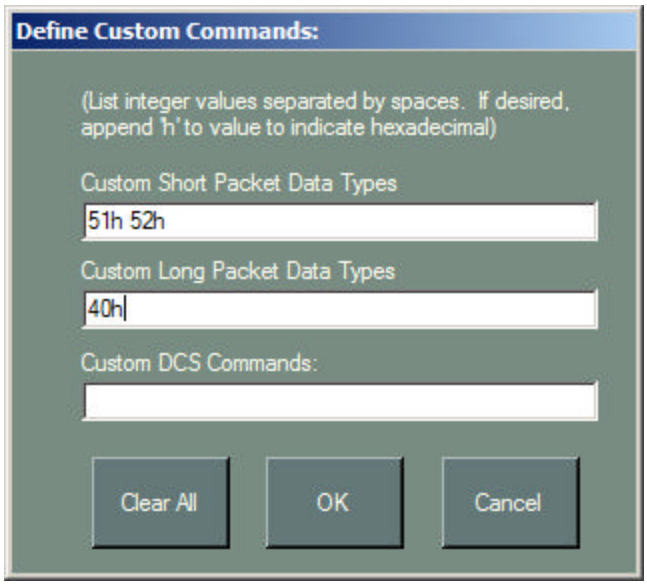

Figure 3 – Custom Command Dialog

Clicking the Cancel button exits the dialog without registering any changes. Clicking the Clear All button clears all the text boxes.

Once custom DataType and DCS commands are been defined, options for selecting these packets will appear in the Custom Commands tree-view node for both triggering and filtering.

4.4 Firmware Update

Periodically, new releases of firmware may be available for the DPhyPreprocessor instrument. Firmware is easily updated through the use of the Firmware dialog, accessible by selecting "Update Firmware" from the Control menu (see Figure 4).

Many of the features of this dialog are for internal use only. For users, the only options are to program new firmware or to verify existing firmware. Simply browse for the firmware filename (extension .rbx) and click either Program or Verify respectively. Once the task has begun, the relevant button changes its lable to Stop, allowing the user to halt the task before completion. This is fine for verification but is not generally recommended when programming, as this will render the instrument inoperative until firmware has been fully programmed.

Either task takes about 5 minutes to complete, with status messages providing updates on progress. When programming is complete, shut down the application and power cycle

the instrument to reload the new firmware. You should check that the first two LEDs on the instrument "twinkle" for a few seconds at power-on, indicating a successful boot. Otherwise, try power cycling again. If the problem persists, you will need to reprogram firmware again.

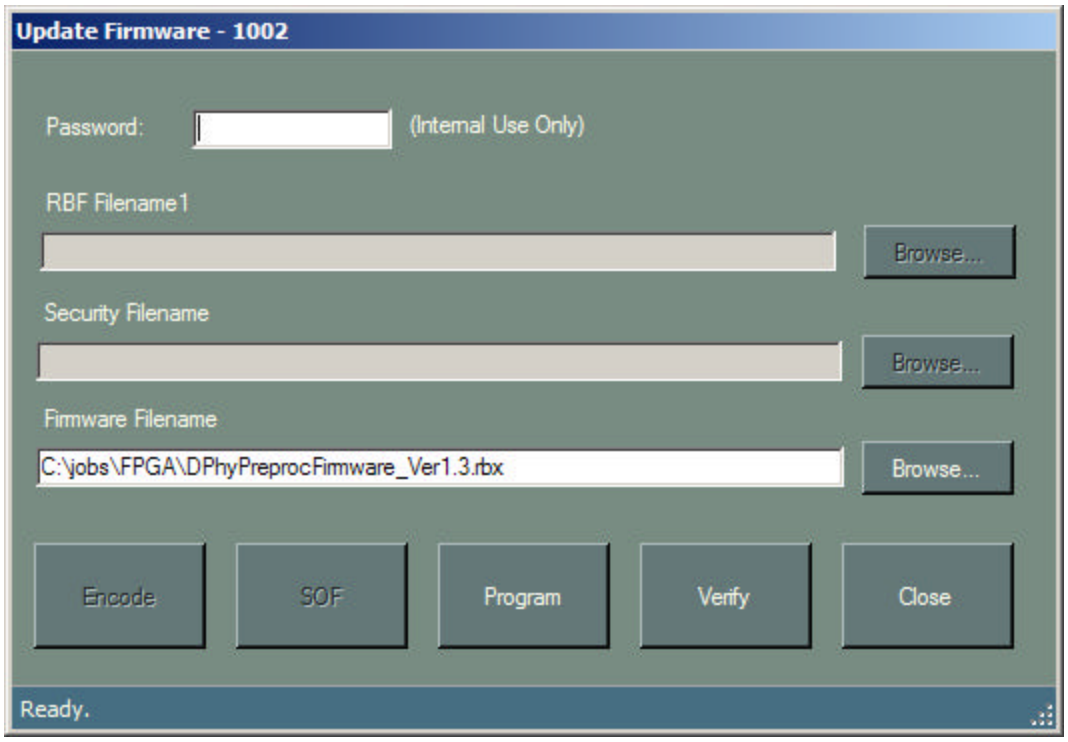

Figure 4 – Firmware Update Dialog

4.5 Configuration Settings

DPhyPreprocCtl saves the current settings when the application is closed and restores these settings upon relaunch. While not generally intended to be edited by the user, the default settings file which is a text file is located at:

C:\Documents and Settings\All Users\Application Data\Moving Pixel Company\DPhyPreprocCtl\DefaultSettings.cfg[5]

User configuration files are created using the File menu save commands. They simply create a named settings file identical to the default settings file that can subsequently be loaded by the user through the File menu load command.

4.6 Menus

This section outlines the menu commands available in DPhyPreprocCtl:

[5] Assuming the application is installed for All Users.

- **File**:
  - **Load –** loads a previously saved command configuration file, overwriting the current configuration.
  - **Save <fn> --** saves the current configuration to a file.
  - **Save <fn> As… --** same as the **Save <fn>** command above except a dialog appears to browse and enter a new configuration file name.
  - **<recent files>** a list of the four most recent configuration files. Selecting one of these file names loads the file.
  - **Clear Recent File List –** clears the most recent file list
  - **Exit** – exits the application
- **Connect**:
  - **DPhy Preprocessor Connect…** – brings up the connection dialog (see section 4.1).
  - **DPhy Preprocessor Disconnect** – disconnects from the instrument, putting the application in offline mode.
  - **Update Firmware**... – brings up a the firmware update dialog to reprogram the DPhy Preprocessor firmware (see section 4.4).
- **Config:**
  - **Define Custom Commands…** – brings up the custom command dialog (see section 4.3)
- **About**
  - **Help** – displays this manual as a help file
  - **About** – brings up a summary window displaying the current software version. This dialog also includes a summary of release notes of changes/fixes for each version.